DTIC FILE COPY

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED Final Report, 1 Aug 89 to 31 Jan 90

4. TITLE AND SUBTITLE
INTERACTIVE GRID GENERATION ON SMALL COMPUTERS

5. FUNDING NUMBERS
F49620-89-C-0096
65502F 3005/A1

2

6. AUTHOR(S)
Peter R. Eiseman

AD-A221 234

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Program Development Corporation
300 Hamilton Avenue, Suite 409
White Plains, NY 10601

8. PERFORMING ORGANIZATION REPORT NUMBER
AFOSR-TR- 90-0505

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NM
Building 410
Bolling AFB, DC 20332-6448

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
F49620-89-C-0096

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release;
distribution unlimited.

12b. DISTRIBUTION CODE
*Original contains color plates: All DTIC reproductions will be in black and white

13. ABSTRACT (Maximum 200 words)

At the start of Phase I, there were several 2D single block codes and a basic mathematical structure for such codes. During the Phase I research, the feasibility for general 2D multiblock codes was firmly established. The mathematical structure of the control point form of algebraic grid generation (CCPF) was found to provide the essential theoretical framework for the codes and accordingly the necessary extensions of the CPF were of central importance. Theses were considered along with the important software techniques and the prototype code. Within the scope of software techniques, the elements of interactive graphics played a central role. The prototype code provided a good test of important ideas.

14. SUBJECT TERMS	15. NUMBER OF PAGES 47
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

Program Development Corporation

300 Hamilton Avenue, Suite 409
White Plains, New York 10601

Tel. 914-761-1732 FAX 914-761-1735

AFOSR

Building 410
Bolling AFB, DC 20332

Phase I Final Report

for contract

F49620-89-C-0096

(unclassified)

Interactive Grid Generation on Small Computers

Submitted by

[illegible]

Peter R. Eiseman, President

(original with original figures)

Abstract

At the start of Phase I, there were several 2D single block codes and a basic mathematical structure for such codes. During the Phase I research, the feasibility for general 2D multiblock codes was firmly established. The mathematical structure of the control point form of algebraic grid generation (**CPF**) was found to provide the essential theoretical framework for the codes and accordingly the necessary extensions of the **CPF** were of central importance. These were considered along with the important software techniques and the prototype code. Within the scope of software techniques, the elements of interactive graphics played a central role. The prototype code provided a good test of important ideas.

Basic Elements for the 2D Multiblock Software

The elements of the **CPF** are the multisurface transformation and the Boolean operations for directional assembly. Accordingly, we shall discuss the multisurface transformation, the assembly, and the extensions. Then the prototype code that was created will be discussed; after which, some graphical images from some of the interactive manipulation will be displayed.

The Multisurface Transformation

The multisurface transformation is based upon an interpolation of tangent vectors to coordinate curves of a given family. The express intent is to guide each such coordinate curve across the field by controlling its tangential direction. Accordingly, the tangent vectors for each curve are assumed to be known at a succession of parameter values that start at the bottom of the parametric interval and end at the top. The interpolation converts the discrete succession of vectors into a continuous field of vectors. Upon integration together with the specified end points, the coordinate curve is obtained.

In the course of the derivation, the discrete tangent vector sequence is seen to come from the successive tangents of a piecewise linear curve that connects the end points. For a sequence of N-1 tangent vectors, there are N-1 piecewise linear links and a sequence of N points that define the piecewise linear curve. When the entire family of coordinate curves is considered, the end points correspond to boundary surfaces and the intermediate points correspond to control surfaces. Because the control is most readily applied in an intuitive manner with the succession of surfaces in space, the transformation was called the multisurface transformation.

Aside from the requirements for transverse smoothness, the basic process can be viewed as indicated here by taking one curve at a time. Moreover, by viewing a succession of points as a succession of 0-dimensional surfaces, they can be used to generate a curve. In this context, the points are most appropriately called control points. In a hiarchial sense, the control points can be used to generate curves which in turn can be used to generate surfaces that in turn can be used to cover volumes, etc. Such hiarchial constructs are employed in the derivation of the **CPF.**

The derivation of the multisurface transformation will be given here in the context of a curve generator from a sequence of control points. This will have the advantage of notational simplicity.without a major loss of content. The general case can be witnessed by mentally replacing the sequence of control points with surfaces.

The curve is assumed to be given by the position vector

$$\mathbf{P} = (x,\ y) \tag{1}$$

which is then given a parameterization by the curvilinear variable ξ. The succession of tangent vectors are given by derivatives $\mathbf{P}'$ at N-1 successive values of ξ. Altogether, the vector field interpolation problem is posed by

$$\mathbf{P}'(\xi) = \sum_{k=1}^{N-1} \psi_k(\xi)\, \mathbf{P}'(\xi_k) \tag{2}$$

where

$$\psi_k(\xi_m) = \delta_{km} \tag{3}$$

for some partition

$$\xi_1 < \xi_2 < \cdots < \xi_{N-1} \tag{4}$$

of the parametric variable. Upon integration, the coordinate positions are given by

$$\mathbf{P}(\xi) = \mathbf{P}_1 + \sum_{k=1}^{N-1} G_k(\xi)\, \mathbf{P}'(\xi_k) \tag{5}$$

where the first term is the constant of integration and where

$$G_k(\xi) = \int_{\xi_1}^{\xi} \psi_k(\sigma)\, d\sigma \tag{6}$$

While the first end point is given by the constant of integration, the second end point is determined by the evaluation

$$\mathbf{P}_N \equiv \mathbf{P}(\xi_{N-1}) = \mathbf{P}_1 + \sum_{k=1}^{N-1} G_k(\xi_{N-1})\, \mathbf{P}'(\xi_k) \tag{7}$$

In succession, the partial sums lead to the sequence of control points that start at the first end point, continue as

$$\begin{aligned} \mathbf{P}_2 &= \mathbf{P}_1 + G_1(\xi_{N-1})\, \mathbf{P}'(\xi_1) \\ \mathbf{P}_3 &= \mathbf{P}_1 + G_1(\xi_{N-1})\, \mathbf{P}'(\xi_1) + G_2(\xi_{N-1})\, \mathbf{P}'(\xi_2) \\ &\vdots \end{aligned} \tag{8}$$

and end at the last end point as given by (7). Since each equation in the sequence that defines successive control points adds only one more term, such terms can be isolated by a subtraction of the previous equation. Since also the isolated terms contain the specified tangents in the form of derivatives, the subtraction gives a conversion of the tangent specification into the form of control points. The consequence is the sequence of N-1 derivative conversions

$$P'(\xi_1) = \frac{1}{G_1(\xi_{N-1})}\{P_2 - P_1\}$$
$$P'(\xi_2) = \frac{1}{G_2(\xi_{N-1})}\{P_3 - P_2\} \qquad (9)$$
$$\vdots$$

With these conversions, the derivatives in (5) are replaced to produce the transformation in terms of control points as given by

$$P(\xi) = P_1 + \sum_{k=1}^{N-1} \frac{G_k(\xi)}{G_k(\xi_{N-1})}\{P_{k+1} - P_k\} \qquad (10)$$

This is called the multisurface transformation. It is determined by the sequence of control points (8), the partition of the parametric coordinate (4), and the interpolation functions (3). The extension from curves to higher dimensional objects is seen, here, as a replacement of the control point sequence by a higher dimensional sequence. That is, controlling objects of dimension n-1 are used to generate objects of dimension n. For control curves, the generation is for a surface grid which, in the 2D plane, is simply a system of coordinates.

The Choice of Interpolation Functions

With the multisurface transformation stated in the general manner of (10), it remains to choose the interpolation functions of (3). There is a wide variety of possibilities for this choice. Namely, any continuous functions that satisfy the Kronecker condition of (3) will work, and moreover, any scaling of such functions is also valid. The arbitrary scaling comes from a direct observation of (10). Each ratio of the G's is a ratio of integrals of the same interpolation function (but over different intervals). Accordingly, any scaling factor would appear in both numerator and denominator; thus, would not affect the transformation.

For precise local control of the curve trajectories, local interpolation functions are preferred. The simplest choice for continuous local functions are the piecewise linear functions. While they are not derivative continuous, their integrals are and accordingly so is the associated multisurface transformation. For convenience, these functions are scaled so that that their integrals over the entire parametric domain is unity. The consequence is that the G's in the denominator of (10) all become 1; thus, simplifying the statement and computation of (10). These interpolation functions are analytically stated as

$$\psi_1(\xi) = \frac{2}{\xi_2 - \xi_1} \begin{Bmatrix} \left(\dfrac{\xi_2 - \xi}{\xi_2 - \xi_1}\right) & \text{for } \xi_1 \le \xi < \xi_2 \\ 0 & \text{otherwise} \end{Bmatrix} \tag{11}$$

about the first partition point, as

$$\psi_k(\xi) = \frac{2}{\xi_{k+1} - \xi_{k-1}} \begin{Bmatrix} 1 - \left(\dfrac{\xi_k - \xi}{\xi_k - \xi_{k-1}}\right) & \text{for } \xi_{k-1} \le \xi < \xi_k \\ 1 - \left(\dfrac{\xi - \xi_k}{\xi_{k+1} - \xi_k}\right) & \text{for } \xi_k \le \xi < \xi_{k+1} \\ 0 & \text{otherwise} \end{Bmatrix} \tag{12}$$

about the k = 2, ... ,N-2 internal partition points, and as

$$\psi_{N-1}(\xi) = \frac{2}{\xi_{N-1} - \xi_{N-2}} \begin{Bmatrix} \left(\dfrac{\xi - \xi_{N-2}}{\xi_{N-1} - \xi_{N-2}}\right) & \text{for } \xi_{N-2} \le \xi \le \xi_{N-1} \\ 0 & \text{otherwise} \end{Bmatrix} \tag{13}$$

about the final partition point.

The Control Point Weighted Format

The statement of the multisurface transformation in (10) gives a close conceptual tie to the process of tangent vector interpolation. While this is certainly important from a fundamental viewpoint, it is not as convenient when repetitive algebraic manipulation is considered. In this latter circumstance, it is preferable to cast it into a control point weighted format. This is done by expanding (10) as

$$P(\xi) = \mathbf{P}_1 + \sum_{k=1}^{N-1} \frac{G_k(\xi)}{G_k(\xi_{N-1})}\{\mathbf{P}_{k+1} - \mathbf{P}_k\}$$

$$= \mathbf{P}_1 + \frac{G_1(\xi)}{G_1(\xi_{N-1})}(\mathbf{P}_2 - \mathbf{P}_1) + \frac{G_2(\xi)}{G_2(\xi_{N-1})}(\mathbf{P}_3 - \mathbf{P}_2) + \cdots + \frac{G_{N-1}(\xi)}{G_{N-1}(\xi_{N-1})}(\mathbf{P}_N - \mathbf{P}_{N-1}) \tag{14}$$

and regrouping as follows

$$P(\xi) = \left\{1 - \frac{G_1(\xi)}{G_1(\xi_{N-1})}\right\}\mathbf{P}_1 + \left\{\frac{G_1(\xi)}{G_1(\xi_{N-1})} - \frac{G_2(\xi)}{G_2(\xi_{N-1})}\right\}\mathbf{P}_2$$

$$+ \cdots + \left\{\frac{G_{N-2}(\xi)}{G_{N-2}(\xi_{N-1})} - \frac{G_{N-1}(\xi)}{G_{N-1}(\xi_{N-1})}\right\}\mathbf{P}_{N-1} + \frac{G_{N-1}(\xi)}{G_{N-1}(\xi_{N-1})}\mathbf{P}_N \tag{15}$$

to get

$$P(\xi) = \sum_{k=1}^{N} \alpha_k(\xi)\,\mathbf{P}_k \tag{16}$$

where the coefficients can be easily identified from (15).

The Domain of Influence

In this format, the centering is about the control points which then each have a domain of influence on the locally surrounding grid points. The center point for each coefficient is its maximum value. With the exception of the end point (k=1 and k=N) maximums, the slopes there are zero. Of the interior coefficients, the ones just adjacent to the end points (k=2 and k=N-1) enter the end points with non zero slope. Each of the remaining interior coefficients (k=3,4, ... , N-3) looks like a bell shaped curve. Each starts at zero slope and zero value at the second partition point on the negative side of the maximum and ends with a zero value and slope at the second partition point on the positive side. Altogether, the coefficient is non zero over three successive intervals determined by the partition points. The maximum point appears in the interior of the interval in the middle. With some modest analysis, the maximum is seen to be located at the parametric position

$$\xi_k^* = \mu\, \xi_k + (1 - \mu)\, \xi_{k-1} \tag{17}$$

where

$$\mu = \frac{\xi_{k+1} - \xi_{k-1}}{(\xi_{k+1} - \xi_{k-1}) + (\xi_k - \xi_{k-2})} \tag{18}$$

between the (k)th and (k+1)st partition points for k=3,4, ... N–3. When k=2 or N–2, half of the bell shaped curve is preserved while the other half is altered. The altered half is on the side of the center which impinges the boundary. The impingement there occurs at a value of zero but with a nonzero slope. The center for these nearly bell shaped coefficients assumes the same form as for the pure bell shapes (17), but with

$$\mu = \frac{\xi_3 - \xi_1}{(\xi_3 - \xi_1) + 2\,(\xi_2 - \xi_1)} \tag{19}$$

and

$$\mu = \frac{\xi_{N-1} - \xi_{N-3}}{(\xi_{N-1} - \xi_{N-3}) + 2\,(\xi_{N-1} - \xi_{N-2})} \tag{20}$$

for k=2 and N-1 respectively. In each case, the center points determined by (17) lie within the interval number k because the blending values in (18), (19), and (20) are strictly between 0 and 1. The remaining coefficients are for the end points. For the first end point, the function strictly decays from a positive value at the end point to zero at the first partition point. For the last end point, the function strictly increases from zero at the next to last partition point to a positive value at the end point.

From the description of the coefficients, the domain over which each is non zero is readily apparent. By including interval end points as well, there are then some points with zero values but there is also a simpler statement that has the essential information. Thus, with this inclusion, the domains are given by the closed sets

$$\text{support}(\alpha_k) = \{\xi : \xi_{k-2} \le \xi \le \xi_{k+1}\} \cap \{\xi \mid \xi_1 \le \xi \le \xi_{N-1}\} \tag{21}$$

for k=1,2, ..., N. The support of a function as in (21) is defined to be the smallest closed set which contains the set of non zero values.

Since the grid on the curve comes from the transformation (16) applied to a uniform grid on the parameterization, the respective supports of the coefficient functions give the respective grid points that are influenced by each coefficient function. Thus, if only one control point is moved from some initial configuration, then the support is used to identify those points which must be moved. Moreover, to identify the most strongly affected point, the center of the corresponding coefficient (17) is computed and the closest grid point (or points, if it is precisely between two) is taken.

Uniformity

Up to this stage, the issues have been placed upon how to generate curves with strong shape control and upon how to ascertain the local extents of such control. The next issue is to address the distribution of points along the generated curve. The main question here is how to determine a uniform distribution of points along the curve. While this could always be done, to some degree, by a subsequent arc length reparameterization; such a reparameterization would acquire non-uniformities in a given direction as the curve experiences continual direction changes. In addition, the reparameterization would require additional work. By identifying the given direction with a vector τ, a consistent measure of uniformity can be established and applied quite simply. This is stated by the projected arc length measure

$$S_P(\xi) = \left\{ \mathbf{P}(\xi) - \mathbf{P}(\xi^*) \right\} \cdot \tau \tag{22}$$

where typically

$$\xi^* = \xi_1 \tag{23}$$

The condition for uniformity is then stated by the requirement that the projected arc length be linear. Using the basic form of the multisurface transformation (10), the projected arc length is given by

$$S_P(\xi) = \sum_{k=1}^{N-1} \frac{G_k(\xi)}{G_k(\xi_{N-1})} d_k \tag{24}$$

in terms of the successive projected distances

$$d_k = \{\mathbf{P}_{k+1} - \mathbf{P}_k\} \cdot \tau \tag{25}$$

between control points. Since the it is the interpolation functions that satisfy a kronecker condition (3) rather than their integrals, there is a motivation to consider the derivative of (24). This will represent a uniform pointwise distribution when it is a constant. It is given by

$$S_p'(\xi) = \sum_{k=1}^{N-1} \frac{\psi_k(\xi)}{G_k(\xi_{N-1})} d_k \tag{26}$$

and, with the kronecker condition, the summation reduces to only one term upon evaluation at the (i)th partition point. Uniformity is satisfied when the derivative equation (26) is set equal to a constant c. The uniformity conditions are then given by

$$c = S_p'(\xi_i) = \frac{\psi_i(\xi_i)}{G_i(\xi_{N-1})} d_i \tag{27}$$

With a substitution from the definition of the interpolation functions in (11), (12), and (13), the integrals (6) in the denominator are 1 and (27) leads to

$$d_i = \frac{c}{2} \left\{ \begin{array}{ll} \xi_2 - \xi_1 & \text{for } i=1 \\ \xi_{i+1} - \xi_{i-1} & \text{for } i=2,3,\cdots,N-2 \\ \xi_{N-1} - \xi_{N-2} & \text{for } i=N-1 \end{array} \right\} \tag{28}$$

The most typical direction to consider is the one which is determined by the end points to the curve. The scaling of this direction will determine the constant c. By setting

$$\tau = (\xi_{N-1} - \xi_1) \frac{\mathbf{P}_N - \mathbf{P}_1}{\| \mathbf{P}_N - \mathbf{P}_1 \|^2} \tag{29}$$

the constant becomes equal to 2 and thereby removes the factor in front of (28). To put the condition on a comparable basis of incremental lengths, it is restated as

$$d_i = \left\{ \begin{array}{ll} h_1 & \text{for } i=1 \\ h_{i-1} + h_i & \text{for } i=2,3,\cdots,N-2 \\ h_{N-2} & \text{for } i=N-1 \end{array} \right\} \tag{30}$$

where the parametric lengths

$$h_i = \xi_{i+1} - \xi_i \tag{31}$$

are balanced against the physical space distances of (25).

When the partition points are chosen to be uniformly distributed, the interpolations functions (11), (12), and (13) at these respective points are simplified to the extent that each interior one is a rigid translation of a single function. Even the boundary functions are represented by the single function with a translation and a doubling of its height. As with any specification of the parametric intervals between the interpolation points, the uniformity condition (30) is addressed as a constraint upon the placement of the control points. The uniform parametric intervals then imply that for each i, the equation (31) has a constant value h. The specific condition from this choice means that the distances (25) are chosen so that the spacing adjacent to the end points (h) *is one half of that in the* interior (2h).

The Attachment of Control Points to a Given Curve

When an arbitrary curve is given in a parametric form, it can be viewed as a transformation from the parameter space to the physical space in which it appears. As a transformation, the objective of attachment is to essentially reproduce it by using the transformation in terms of control points (16). This amounts to an approximation of both its geometry and pointwise distribution. The accuracy in this process is usually quite good for a modest number of control points and is, of course, enhanced as the number of selected control points N increases.

In terms of the pointwise distribution, the starting condition is with a uniform distribution in parameter space. Any non-uniformities along the curve are then the result of applying the given map to the given curve. Accordingly, if the map is to be reproduced, then the uniform parameter space must first be reproducible. Given any number N of control points, this can be accomplished by placing the control points in the parameter space in such a manner as to satisfy the **uniformity** condition above. In the case with equally distributed interpolation points, the N points determine N-1 lengths of equal size; thus, N-2 interior lengths of equal size and a half length for each end by splitting the remaining one. If the interior length is 2h, then the lengths for the ends is h where

$$h = \frac{\xi_{N-1} - \xi_1}{2(N-1)} \tag{32}$$

Once the control points are placed in parameter space, the most immediate approximation is then to send those control points into physical space by applying the given curve defining map. If the curve is a uniform straight line segment, then the attachment should be exact since the projected arc length is along the actual curve and since the distribution is linear. That is for a given curve

$$\gamma(\xi) = \left(1 - \frac{\xi - \xi_1}{\xi_{N-1} - \xi_1}\right)\mathbf{a} + \left(\frac{\xi - \xi_1}{\xi_{N-1} - \xi_1}\right)\mathbf{b} \tag{33}$$

we get

$$\mathbf{P}(\xi) = \gamma(\xi) \tag{34}$$

for the transformation **P** in (16) when

$$\begin{aligned}
\mathbf{P}_1 &= \gamma(\xi_1) = \mathbf{a} \\
\mathbf{P}_2 &= \gamma(\xi_1 + h) \\
\mathbf{P}_3 &= \gamma(\xi_1 + 3h) \\
&\vdots \\
\mathbf{P}_{N-1} &= \gamma(\xi_1 + 2(N-2)h) \\
\mathbf{P}_N &= \gamma(\xi_1 + 2(N-1)h) = \gamma(\xi_{N-1}) = \mathbf{b}
\end{aligned} \tag{35}$$

When the linear curve of (33) is replaced by an arbitrary continuous curve, the control point sequence of (35) leads to a transformation (16) that is an approximation to the given curve γ.

The approximation is typically quite close for even modest numbers of control points N.

The error of approximation is due to the effect of curvature. This occurs because the control point determination by mapping (35) does not account for the direction changes in the curve. To witness the effect, the curve can be viewed as a succession of curved segments that are separated by inflection points about which the changes in concavity appear. That is the separation is into segments with non zero curvature. If, for example, the approximation is considered within such a segment, then the control point curve is seen to lie on the concave side of the given curve. To account for curvature in this case, the control points must be lifted off of the given curve and displaced in the outward convex direction in an amount that is just sufficient to drag the associated control point curve closer to the given curve. The measure of closeness can be given in a number of standard norms and various strategies are possible for an associated optimization process. These procedures can also be expected to extend to sections including inflection points.

Up to this stage in the discussion, the given curve is represented in an analytical form. In practice, the more general approach is to assume the curve γ to be given in the discrete form of a grid: namely, it is presented as a sequence of points in physical space. Without loss of generality, the sequential index can be taken as the parameterization. As such, it is then viewed as a mapping from index space into physical space. With the indices being mapped onto their corresponding grid points, the parameter values between indices are most simply defined to be mapped in the local linear fashion using the closest index locations. These are determined by directly finding the greatest integer part of the parameter value, taking the amount beyond it, and using that amount for the proportionate distance in the linear interpolation between associated grid points.

The generality in the grid representation is readily seen since any analytically given curve γ can be converted to a grid by the mapping of uniformly spaced parametric points that cover the parametric domain. This grid and its associated piecewise linear curve represent another approximation to γ. Accordingly, there is some associated error. When the attachment process as given by (35) is applied to determine a control point transformation, the total error involved is a composite of both the error from control point placement on γ and the conversion from analytic to discrete.

In the prototype code, the attachment process was executed in the manner described by the pure mapping strategy as applied to curves and two dimensional grids in the multiblock structure. Altogether, this is a good first step. However, it is not the end of the story. Quite simply, the process must be improved to include the effects of curvature.and to generally enhance the accuracy. This suggested improvement has been reserved for Phase II. The benefit will be seen most directly for 2D grids when strong clustering is required at a specified boundary that is highly curved. Without this enhancement, the attachment by mapping only will lead to an overlapped grid. A situation that demands strong clustering in the presence of high curvature can be expected to commonly arise in the solution of the Navier-Stokes equations.

The Basic Transfinite Interpolation Process

The generation of a curve that connects two end points is an example of a uni-directional construct: the direction of construction being only along the curve. When such curves are assembled to form a family of coordinate curves, the construct is still a uni-directional one. The directional parameter is identified by the curvilinear coordinate variable that is the parameterization for each coordinate curve in the family. The unidirectional constructs are characterized by their very nature in which the only real specifications of data occur at fixed stations of the corresponding parameter value. In the simplest case of straight line constructs, the specification is just the end point positions in space. When these straight lines are smoothly assembled to form a coordinate system, the end point specifications become curve or surface specifications. The result is called a **shearing transformation**. The clear defect in this process is the fact that there are no real specifications in the remaining coordinate directions that were inherited from the specified boundaries. To correct this defect, a further process is needed to assemble these directional constructs in such a manner as to conform to specifications in more than one direction. This process has been called **transfinite interpolation**.

Transfinite interpolation is most simply described by demonstrating it for shearing transformations. In two dimensions, there are two shearing transformations that can be stated as

$$\mathbf{A}(\xi,\eta) = (1-\xi)\,\mathbf{P}(0,\eta) + \xi\;\mathbf{P}(1,\eta) \tag{36}$$

and

$$\mathbf{B}(\xi,\eta) = (1-\eta)\,\mathbf{P}(\xi,0) + \eta\;\mathbf{P}(\xi,1) \tag{37}$$

for the respective ξ and η coordinate directions which, for simplicity, are taken over the unit domain

$$\begin{aligned} &0 \le \xi \le 1 \\ &0 \le \eta \le 1 \end{aligned} \tag{38}$$

The fixed stations for (36) are for ξ equal to 0 and 1 and are represented by the respective specified boundaries

$$\mathbf{P}(0,\eta) \quad \text{and} \quad \mathbf{P}(1,\eta) \tag{39}$$

However, when η is equal to 0 and 1, the shearing (36) gives straight lines rather than the more general specified boundaries.

$$\mathbf{P}(\xi,0) \quad \text{and} \quad \mathbf{P}(\xi,1) \tag{40}$$

With the shearing in the η-direction (37), the specifications are reversed. The specified boundaries are given by (40) while the lateral boundaries are given by straight lines rather than (39). Since the lateral boundaries of the shearings (36) and (37) together represent a linear rendition of each boundary, the natural inclination is to develop the transformation which produces linearity at each boundary. Clearly, the corner points are all that is required. These represent the intersection of the edges. This transformation is given by the double shearing: namely, to shear with (36) between the corners defined by η equal to 0 and 1 and then to use these straight line results in place of (40) in the shearing of (37). The order of the double application of shearings can be reversed by employing (37) first and then (36). The result is the same. It is called the **tensor product** transformation and is given by

$$T(\xi,\eta) = (1-\xi)(1-\eta)\mathbf{P}(0,0) + \xi(1-\eta)\mathbf{P}(1,0) + (1-\xi)\eta\mathbf{P}(0,1) + \xi\eta\mathbf{P}(1,1) \tag{41}$$

where the corner points are explicitly blended into each other. The specification of all boundaries, however, requires an interpolation of the union of edges rather than the intersection as given by the tensor product. By returning to the directional shearings of (36) and (37), their vector sum (**A**+**B**) contains the fixed stations for boundary specifications at all boundaries, but also contains line segments as well. That is, the evaluation at the boundaries reduces to the vector sum of the position vector for boundary specification and the position vector for the straight line segment that connects the corners corresponding to the end points. Since these straight line segments appear at each boundary and are also produced by the tensor product transformation **T** of (41), the transformation which conforms to all boundaries is obtained by a subtraction of **T** from the sum **A+B** and is given by

$$\mathbf{P} = \mathbf{A} + \mathbf{B} - \mathbf{T} \tag{42}$$

which defines the position vector **P** everywhere throughout the domain (38) from its specification along the entire domain boundary. of (38). Because this represents an interpolation of the entire boundary which can generally be given as a continuum which then involves an infinite number of points, it has been called a **transfinite interpolation** to reflect the possible infinity of points in comparison to the finite set represented by the corners for the tensor product. In terms of the boundary point specifications, the tensor product (41) and transfinite transformations (42) respectively interpolate the edge intersections and unions. In a more general manner, the shearing transformations can be viewed as separate projections from the class of all transformations that conform to the opposing boundaries to the specific choice of the shearing. This requires some extra notation to accommodate the projectors, but the essential features follow the same pattern as given here. In the notation of projectors, the tensor product transformation (41) and the transfinite transformation (42) are called the **Boolean** product and sum respectively. The use of projectors extends well beyond the specific choices of assembling shearings. The general feature with union and intersections still holds. The Boolean products and sums respectively correspond to conformity with the intersections and unions of the associated objects. Those objects, in our illustration were only the edges. This general format provides the framework to proceed with the assembly of multisurface transformations.

The Basic Control Point Form of Algebraic Grid Generation

The control point form of algebraic grid generation is given here first to delineate the basic structure from which the more general bottoms-up self consistent formulation will be presented for multiblock application. The basic **CPF** is obtained from the Boolean assembly of multisurface transformations. The assembly is done in such a manner that the control rests with a sparse array of control points rather than an intersecting network of complete control surfaces. The latter would arise from the rather straight forward application of the transfinite procedures. It would also present the burden of more data to manipulate as well as other problems.

The constructive elements of the **CPF** are the control point array, the interpolation points, the associated blending functions, the specified boundaries, and the switches for boundary specification. With the exception of the last stated element, these are now to be given a notational designation. The control point array is given by

$$\{ \ Q_{ij} \ \mid \ i = 1,2,\cdots,I \quad \text{and} \quad j = 1,2,\cdots,J \ \} \tag{43}$$

The vector field interpolation points (4) for the multisurface transformation (10) are given by the partitions points

$$\begin{gathered} \xi_1 < \xi_2 < \cdots < \xi_{I-1} \\ \eta_1 < \eta_2 < \cdots < \eta_{J-1} \end{gathered} \tag{44}$$

the associated blending functions (16) are designated by

$$\begin{aligned} &\alpha_i(\xi) && \text{for } i = 1, 2, \cdots, I \\ &\beta_j(\eta) && \text{for } j = 1, 2, \cdots, J \end{aligned} \tag{45}$$

and the specified boundaries are stated in previous manner of (39) and (40) as

$$\mathbf{P}(\xi,\eta_1) \qquad \mathbf{P}(\xi,\eta_{J-1}) \qquad \text{for} \qquad \xi_1 \le \xi \le \xi_{I-1}$$

$$\mathbf{P}(\xi_1,\eta) \qquad \mathbf{P}(\xi_{I-1},\eta) \qquad \text{for} \qquad \eta_1 \le \eta \le \eta_{J-1} \tag{46}$$

The construction starts with the creation of a control point curve (16) for each row and column of the control point array (43). These curves will lead to the tensor product **T** and to the directional transformations that reflect the **A** and **B** in (36) and (37). These are generated as

$$\mathbf{a}_j(\xi) = \sum_{i=1}^{I} \alpha_i(\xi)\,\mathbf{Q}_{ij} \tag{47}$$

for j = 1, 2, ... , J and

$$\mathbf{b}_i(\xi) = \sum_{j=1}^{J} \beta_j(\xi)\,\mathbf{Q}_{ij} \tag{48}$$

for i = 1, 2, ... , I. The tensor product is computed by using the curves in (48) for constructive surfaces as

$$\mathbf{T}(\xi,\eta) = \sum_{i=1}^{I} \alpha_i(\xi)\,\mathbf{b}_i(\eta) \tag{49}$$

or, alternatively, the curves of (47) as

$$\mathbf{T}(\xi,\eta) = \sum_{j=1}^{J} \beta_j(\eta)\,\mathbf{a}_j(\xi) \tag{50}$$

which each reduce to the symmetric formula

$$\mathbf{T}(\xi,\eta) = \sum_{i=1}^{I}\sum_{j=1}^{J} \alpha_i(\xi)\,\beta_j(\eta)\,\mathbf{Q}_{ij} \tag{51}$$

that shows the invariance to the order of assembly: that is, the commutativity of ;this process. By viewing the tensor product as coming from the constructive surfaces of (47) as in (50) and of (48) as in (49), the evaluations at the boundaries are readily seen to be given by

$$\mathbf{T}(\xi_1,\eta) = \mathbf{b}_1(\eta) \qquad \mathbf{T}(\xi_{I-1},\eta) = \mathbf{b}_I(\eta)$$
$$\mathbf{T}(\xi,\eta_1) = \mathbf{a}_1(\xi) \qquad \mathbf{T}(\xi,\eta_{J-1}) = \mathbf{a}_J(\xi) \tag{52}$$

since each multisurface construct matches its boundary (5), (7). This result simply states that the tensor product transformation reduces to a control point curve at each boundary

The tensor product transformation, like its simpler counterpart in the case of shearing transformations in (41), does not utilize the data for specified boundary edges: it is only dependent upon the control point array (43). It now remains to develop the directional constructs which include the missing boundary data and correspond to the previous **A** and **B** of (36) and (37). This is obtained by adjusting the multisurface assembly of control point curves to get the tensor product. The adjustment comes from a replacement of control point curve boundaries with the associated specified boundaries. The pertinent modification of (49) then becomes

$$\mathbf{A}(\xi,\eta) = \alpha_1(\xi)\,\mathbf{P}(\xi_1,\eta) + \sum_{i=2}^{I-1} \alpha_i(\xi)\,\mathbf{b}_i(\eta) + \alpha_I(\xi)\,\mathbf{P}(\xi_{I-1},\eta) \tag{53}$$

and, similarly, the other direction (50) becomes

$$\mathbf{B}(\xi,\eta) = \beta_1(\xi)\,\mathbf{P}(\xi,\eta_1) + \sum_{j=2}^{J-1} \beta_j(\eta)\,\mathbf{a}_j(\xi) + \beta_J(\eta)\,\mathbf{P}(\xi,\eta_{J-1}) \tag{54}$$

Because the intermediate control curves are, by construction, the control point curves used to develop the tensor product in their respective directions, it is reasonable to complete the corresponding summation in the middle of (53) and (54) to cover the full range so that it becomes the tensor product. In so doing, the boundary control point curves are added into the summation and are balanced by their subtraction from the first and last terms on each end. The result is a tensor product for the center term and the remaining terms being the boundary blending terms. In the first direction, the transformation of (53) becomes

$$\mathbf{A}(\xi,\eta) = \mathbf{T}(\xi,\eta) + \alpha_1(\xi)\left\{\mathbf{P}(\xi_1,\eta) - \mathbf{b}_1(\eta)\right\} + \alpha_I(\xi)\left\{\mathbf{P}(\xi_{I-1},\eta) - \mathbf{b}_I(\eta)\right\} \tag{55}$$

where the last two terms are the boundary blending terms. Similarly, the transformation in the other direction becomes

$$\mathbf{B}(\xi,\eta) = \mathbf{T}(\xi,\eta) + \beta_1(\eta)\left\{\mathbf{P}(\xi,\eta_1) - \mathbf{a}_1(\xi)\right\} + \beta_J(\eta)\left\{\mathbf{P}(\xi,\eta_{J-1}) - \mathbf{a}_J(\xi)\right\} \qquad (56)$$

Altogether the boundary adjustment terms appear as a difference between the specified boundary **P** and a control point curve which is scaled by the multisurface blending function at the corresponding boundary. Since the control point curve at each boundary is determined by the corresponding boundary control points from the array (43) and since that represents just an evaluation of the tensor product **T**, these directional multisurface constructs can be expressed entirely in terms of the specified boundary and the tensor product. That is using (52), the directional constructs (55) and (56) are now written as

$$\mathbf{A}(\xi,\eta) = \mathbf{T}(\xi,\eta) + \alpha_1(\xi)\,\mathbf{H}(\xi_1,\eta) + \alpha_I(\xi)\,\mathbf{H}(\xi_{I-1},\eta) \qquad (57)$$

and

$$\mathbf{B}(\xi,\eta) = \mathbf{T}(\xi,\eta) + \beta_1(\eta)\,\mathbf{H}(\xi,\eta_1) + \beta_J(\eta)\,\mathbf{H}(\xi,\eta_{J-1}) \qquad (58)$$

respectively where

$$\mathbf{H} = \mathbf{P} - \mathbf{T} \qquad (59)$$

is the is the difference between the general position vector and the tensor product which, here, is evaluated at only the four boundaries. With these directional constructs and the tensor product, the next step is to assemble the pieces in the transfinite manner (42) of

$$\mathbf{P}(\xi,\eta) = \mathbf{A}(\xi,\eta) + \mathbf{B}(\xi,\eta) - \mathbf{T}(\xi,\eta) \qquad (60)$$

This yields the basic control point transformation and is given by

$$\begin{aligned}\mathbf{P}(\xi,\eta) = \mathbf{T}(\xi,\eta) &+ \rho_1\,\alpha_1(\xi)\,\mathbf{H}(\xi_1,\eta) + \rho_2\,\alpha_I(\xi)\,\mathbf{H}(\xi_{I-1},\eta)\\ &+ \rho_3\,\beta_1(\eta)\,\mathbf{H}(\xi,\eta_1) + \rho_4\,\beta_J(\eta)\,\mathbf{H}(\xi,\eta_{J-1})\end{aligned} \qquad (61)$$

where an extra coefficient has been added to each boundary adjustment term. These are denoted by ρ_k for k=1,2,3,4. and act as switches which turn each boundary blending term on or off in correspondence with the values of 1 and 0 respectively.

The derivation, here, has deductively lead to the form with the values of 1. Upon inspection of the deductive result, the interpretation of the boundary adjustment terms has lead to the generalization to include the switches. The interpretation is that the each adjustment term acts individually for a particular boundary. The other boundaries and the tensor product core remain the same. Thus, by dropping an adjustment term, the blending from the specified to the control point representation is simply removed; thus, leaving only the tensor product for the given boundary which ,of course, is only determined by the boundary control points.

The Bottoms-Up Representation For The Topological Stencil

In the basic control point form of the previous section, the specified boundaries appear as arbitrarily given entities. There are simply no restrictions as to how those boundary grid points were generated. While this is certainly a general statement, there is also *no flexible means offered to model such boundaries.* That modeling issue is addressed in this section. Accordingly, the control point form of algebraic grid generation will now be extended to include the boundary modeling capability. It is done in a self-consistent manner which is a very flexible bottoms-up approach. The self consistency comes from the use of the same mathematical machinery for the purpose of boundary modeling. The bottoms-up philosophy connotes that we start from the simpler objects and build up to the more complex ones. Thus, from points, the curves including boundaries are generated, and then from such curves, the two dimensional coordinates are obtained. In addition to the geometry modeling, the benefits of this form include the format for actions that utilize storage compactness and a ready means to perform the assembly of coordinate patches into a multiple block configuration.

With the use of the same mathematical structure for the boundary grids, the notation for the associated curves is most conveniently taken to parallel that of the development for the two dimensional grid patch. In the i-direction, the two boundary curves are distinguished by k=1,2 and are given by

$$\mathbf{u}^k(\xi) = \sum_{i=1}^{I_k} \alpha_i^k(\xi)\, \mathbf{r}_i^k \tag{62}$$

for control points

$$\left\{ \mathbf{r}_i^k \;\mid\; \text{for } i = 1, 2, \cdots, I_k \right\} \tag{63}$$

and blending functions $\alpha_i^k(\xi)$ as in (16) defined over the partition of multisurface interpolation points

$$\xi_1 = \xi_1^k < \xi_2^k < \cdots < \xi_{I_k-1}^k = \xi_{I-1} \tag{64}$$

where the parametric range matches that for the area grid generation, but has a generally larger number of points. The disparity in the number of control points occurs because the boundary modeling is typically more detailed than is required within the area grid. Thus, the richer supply of control points can be brought to bear upon the more rigid boundary requirements without having to be propagated into the area. With similar considerations the boundaries in the j-direction are given as

$$\mathbf{v}^k(\eta) = \sum_{j=1}^{J_k} \beta_j^k(\eta)\, \mathbf{s}_j^k \tag{65}$$

for control points

$$\left\{ \mathbf{s}_j^k \;\mid\; \text{for } j = 1, 2, \cdots, J_k \right\} \tag{66}$$

and partition points

$$\eta_1 = \eta_1^k < \eta_2^k < \cdots < \eta_{J_k-1}^k = \eta_{J-1} \tag{67}$$

with blending functions $\beta_j^k(\eta)$ of the form in (16). With the boundary specification given by

$$\begin{aligned} \mathbf{P}(\xi,\eta_1) &= \mathbf{u}^1(\xi) \qquad & \mathbf{P}(\xi,\eta_{J-1}) &= \mathbf{u}^2(\xi) \\ \mathbf{P}(\xi_1,\eta) &= \mathbf{v}^1(\eta) \qquad & \mathbf{P}(\xi_{I-1},\eta) &= \mathbf{v}^2(\eta) \end{aligned} \tag{68}$$

the self-consistent control point form of algebraic grid generation is given by

$$\begin{aligned} \mathbf{P}(\xi,\eta) = \mathbf{T}(\xi,\eta) &+ \rho_1\, \alpha_1(\xi) \left\{\mathbf{v}^1(\eta) - \mathbf{T}(\xi_1,\eta)\right\} \\ &+ \rho_2\, \alpha_I(\xi) \left\{\mathbf{v}^2(\eta) - \mathbf{T}(\xi_{I-1},\eta)\right\} \\ &+ \rho_3\, \beta_1(\eta) \left\{\mathbf{u}^1(\xi) - \mathbf{T}(\xi,\eta_1)\right\} \\ &+ \rho_4\, \beta_1(\eta) \left\{\mathbf{u}^2(\xi) - \mathbf{T}(\xi,\eta_{J-1})\right\} \end{aligned} \qquad (69)$$

in the bottoms-up manner.

The topological stencil in the prototype 2D multiblock code was given in terms of the edge control points. This was established by first having the user specify vertices from which the starting edges were generated by lines such as (33). Then the attachment process was applied to produce initial edge control points. The user was only required to specify the desired number of control points. Then from the initial control points the action shifted to their manipulation by movement so that the boundary edges can be modeled into the desired shapes. From the edges, the prototype code was then structured to generate the area grids for each coordinate system in the topological *template* of edges. Altogether, the structure follows the mathematical formulation derived here for this purpose.

The Computational Speed

The application of the control point form of algebraic grid generation can be executed in a particularly rapid manner during the process of interactive grid manipulation. The speed, here, is substantially greater than that of a straight forward application. The process by which the computational speed is improved for the dynamic manipulation process rests upon the localization of the computations and the computation of only the changes in the grid. This latter aspect is considered first. The starting condition was obtained by generating the initial control point grid with the optimization to the extent of utilizing the local analytic structure While this is faster than the straight forward computation, it is still not as fast as the process which is based upon the computation of changes. Fortunately, the requirements for speed at the initialization stage are not as great as that during the dynamic manipulation stages which comprise the bulk of the interactive use of the code. .

Given the initial starting grid together with its control net, the main subsequent action comes with the change of position of a control point or group of control points. Accordingly, it is reasonable to compute only the changes in the grid point positions that are influenced by the altered control points. While this represents a clear savings that can be seen from the supports of the control point coefficients as stated in (21), a further savings was obtained by employing the previous known grid.

The strategy was to compute the difference between the new and old grids and then to add this computed difference or "delta form" to the old grid. This will be discussed first for the **area** control points and then for the **edge** control points in the bottoms up formulation

The Delta Form For Area Control Points

For a change of position in one of the area control points (43), the delta form can be computed with either the **CPF** stated in (61) or in (69). In either case, the boundary specifications cancel out with the result that is expressed as

$$\begin{aligned}\Delta\mathbf{P}(\xi,\eta) = \Delta\mathbf{T}(\xi,\eta) - \rho_1\,\alpha_1(\xi)\;\Delta\mathbf{T}(\xi_1,\eta) - \rho_2\,\alpha_I(\xi)\;\Delta\mathbf{T}(\xi_{I-1},\eta)\\ - \rho_3\,\beta_1(\eta)\;\Delta\mathbf{T}(\xi,\eta_1) - \rho_4\,\beta_J(\eta)\;\Delta\mathbf{T}(\xi,\eta_{J-1})\end{aligned} \tag{70}$$

where

$$\Delta\mathbf{P} = \mathbf{P}^{new} - \mathbf{P}^{old} \tag{71}$$

and

$$\Delta\mathbf{T} = \mathbf{T}^{new} - \mathbf{T}^{old} \tag{72}$$

Of the area control points, there are three cases to consider for movement. They are the respective motion of corner, edge, and interior control points. The corners will be examined first. It is sufficient to look at the action for a typical corner. Thus, consider the motion of the corner with i=1 and j=1. If only that point is moved, then its delta is non-zero which is stated as

$$\Delta\mathbf{Q}_{ij} = 0 \;\text{ when }\; i \neq 1 \text{ and } j \neq 1 \tag{73}$$

but

$$\Delta \mathbf{Q}_{11} \neq 0 \tag{74}$$

where

$$\Delta \mathbf{Q}_{ij} = \mathbf{Q}_{ij}^{new} - \mathbf{Q}_{ij}^{old} \tag{75}$$

Next, the general delta form of (70) must be evaluated. Clearly, the boundary terms which do not involve the (1,1)-term disappear. This removes the second and fourth such terms. The remaining tensor product delta's involve the change of (74). These are computed from (51) as

$$\begin{aligned} \Delta T(\xi,\eta) &= \alpha_1(\xi)\, \beta_1(\eta)\, \Delta Q_{11} \\ \Delta T(\xi_1,\eta) &= \alpha_1(\xi_1)\, \beta_1(\eta)\, \Delta Q_{11} = \beta_1(\eta)\, \Delta Q_{11} \\ \Delta T(\xi,\eta_1) &= \alpha_1(\xi)\, \beta_1(\eta_1)\, \Delta Q_{11} = \alpha_1(\xi)\, \Delta Q_{11} \end{aligned} \tag{76}$$

When these delta's are inserted into the general delta form of (70), the form reduces to

$$\Delta \mathbf{P} = (1 - \rho_1 - \rho_3)\, \alpha_1(\xi)\, \beta_1(\eta)\, \Delta \mathbf{Q}_{11} \tag{77}$$

If either both of the boundaries that contain the common corner are considered to be specified, then both of the switches are on and the corresponding values of ρ_1 and ρ_3 are unity. The delta in this case is negative in correspondence to one plus from the tensor product core and two negatives from the boundary adjustment terms. When one boundary is considered to be specified and the other to be determined by only area control points, then one switch in 0 and the other is 1. The result is then a direct balance of the tensor product core versus the one alive adjustment term. This leads to no movement at all! : the grid is invariant. The last possibility is for both boundaries to be considered to be determined by only area control points. In this instance, there are no adjustment terms for boundaries and accordingly, as expected the delta comes entirely from the core and is a positive contribution. To get the new grid positions, the computed delta is added to the old grid point positions according to the formula

$$\mathbf{P}^{new} = \mathbf{P}^{old} + \Delta \mathbf{P} \tag{78}$$

For this corner point motion, the computation is localized to the grid points influence by the new corner point position. The effected points are determined by the domain of influence of the coefficients. With the support as given in (21), the effected grid points in this instance are those which correspond to the curvilinear section of the blending support given by

$$\xi_1 \le \xi < \xi_2$$
$$\eta_1 \le \eta < \eta_2 \qquad (79)$$

Thus, the delta is non-zero only in this subdomain and, as a result, the grid remains unchanged outside of this local region: that is the application of (78) is only a very local one.

While the corner points belong to two edges and technically can be considered as edges points, the edge points for this discussion are purely internal to the specific edge under consideration. Accordingly, the area control points which lie on edges lead to a delta form that has only the boundary adjustment term belonging to the single specific corresponding edge For example, if only the area control point in the (1,2)-index location is moved, then the delta form of (70) reduces to

$$\Delta\mathbf{P}(\xi,\eta) = \Delta\mathbf{T}(\xi,\eta) - \rho_1\, \alpha_1(\xi)\, \Delta\mathbf{T}(\xi_1,\eta) \qquad (80)$$

With

$$\Delta\mathbf{T}(\xi,\eta) = \alpha_1(\xi)\, \beta_2(\eta)\, \Delta\mathbf{Q}_{12} \quad \text{and} \quad \Delta\mathbf{T}(\xi_1,\eta) = \beta_2(\eta)\, \Delta\mathbf{Q}_{12} \qquad (81)$$

the delta becomes

$$\Delta\mathbf{P} = (1 - \rho_1)\, \alpha_1(\xi)\, \beta_2(\eta)\, \Delta\mathbf{Q}_{12} \qquad (82)$$

to give the new position

$$\mathbf{P}^{new} = \mathbf{P}^{old} + \Delta\mathbf{P} \qquad (83)$$

If the boundary is considered to be specified, then the delta is zero because its leading coefficient (with the switch turned on) vanishes. In effect, the tensor product transformation is experiencing the change but that change is not appearing in the **CPF** because the boundary adjustment action provides a precise balance which annihilates it. That is, the **CPF** is invariant while the tensor product is changed by an arbitrary amount. This arbitrary change is seen when the boundary is not specified for then there is no balancing adjustment term and the action is solely that of the tensor product core. The leading coefficient in (82) is then unity.

The remaining possibility to consider for area control point motion is the more standard motion when the moved point is completely interior . That is, it is not on any boundary, be it a corner or an edge. In this case, the boundary adjustment terms do not appear since the associated deltas depend solely upon the control points along the boundaries. Thus, the delta form comes from the tensor product core, and in particular, from the term in (51) which has been changed. Accordingly, when only the control point at the internal index location (i,j) is moved, the new position becomes

$$\mathbf{P}^{new} = \mathbf{P}^{old} + \alpha_i(\xi)\,\beta_j(\eta)\,\Delta\mathbf{Q}_{ij} \tag{84}$$

The Delta Form For Geometry Modeling Control Points

The next consideration is for the motion of the control points that are associated with the boundaries in the self-consistent bottoms-up formulation. These are the geometry modeling points. With the assumption that the motion is restricted to only the geometry modeling points, the delta form is computed as in (70) from a difference of CPF's; but this time, the CPF is restricted to the self-consistent one of (69). In contrast to the situation with the area control points, the action is totally contained within the adjustment terms; and moreover, within those terms, it is the tensor product which is cancelled out in the delta between new and old. This also changes the sign of each adjustment contribution since it is a delta between positive rather than negative terms which contribute. To witness the effect in concrete terms, it is sufficient to examine the movement of a single specific point. Thus, if the third modeling point on the third boundary of (69) is moved to a new location, then the delta is given by

$$\begin{aligned}\Delta\mathbf{P} &= \rho_3\,\beta_1(\eta)\left\{\Delta\mathbf{u}^1(\xi)\right\}\\ &= \rho_3\,\beta_1(\eta)\left\{\alpha_3^1(\xi)\Delta\mathbf{r}_3^1\right\}\end{aligned} \tag{85}$$

where the modeling control point is explicitly given from using (62). The new position is given by

$$\mathbf{P}^{new} = \mathbf{P}^{old} + \alpha_3^1(\xi)\,\beta_1(\eta)\,\Delta\mathbf{r}_3^1 \tag{86}$$

where the switching factor was taken to be unity since the boundary motion has an effect only if the boundary is considered as specified.

The Motion for Groups of Control Points

With the motion described for individual points that are associated with various parts of the area or boundary modeling control, the motion for groups of control points is the next practical consideration. The simplest instance of such group motions are those cases where several points are tied together in the sense of a common usage. In the case of boundary modeling points, the end points of curves lead to the natural inclination of uniquely defining corner points. For example, in the (1,1) location, the first and third curves should come together. In the spirit of (86), there are two delta contributions that together lead to

$$\begin{aligned}\mathbf{P}^{new} &= \mathbf{P}^{old} + \alpha_1^1(\xi)\,\beta_1(\eta)\,\Delta\mathbf{r}_1^1 + \alpha_1(\xi)\,\beta_1^1(\eta)\,\Delta\mathbf{s}_1^1 \\ &= \mathbf{P}^{old} + \left[\alpha_1^1(\xi)\,\beta_1(\eta) + \alpha_1(\xi)\,\beta_1^1(\eta)\right]\Delta\mathbf{r}_1^1\end{aligned} \tag{87}$$

where the last equation represents the identification of the corner. Moreover, when the area control point for the same corner is also identified, yet another adjustment term must be included. This, of course, is quite natural to consider and represents a transition from the uniqueness with respect to only edges to a uniqueness that is for the entire **CPF**. With (87) and (77), the unique corner motion is then given by

$$\mathbf{P}^{new} = \mathbf{P}^{old} + \left[\alpha_1^1(\xi)\,\beta_1(\eta) + \alpha_1(\xi)\,\beta_1^1(\eta) - \alpha_1(\xi)\,\beta_1(\eta)\right]\Delta\mathbf{Q}_{11} \tag{88}$$

In the event that an edge modeling point can be identified with an edge control point for the area, the possible joint movement can be considered. In so doing, however, the grid is determined in the direct manner of a pure motion of the modeling point as in (86). The contribution of the area control point is zero because it vanishes as in (82) when the boundary is considered to be specified.

Up to this stage, the various special cases have been examined. These included the corner, edge, and internal points for the control net over the area; the edge and corner points for the separate control nets over the boundary curves; and the instances when there would be possible coincident points among these nets. In the special cases, the distinct actions were assembled in an additive manner. This was possible because there is linearity with respect to the control nets. In a larger sense, the linearity leads to a general **superposition principle** whereby the motion of a group of control points are viewed as a sum of individual changes that are added together to obtain the change for the group. Accordingly, with the knowledge of the various possible special cases, the general motion of groups of points is then determined by the superposition of the associated deltas for the special cases.

The Number of Computed Grid Points

Once the group of control points to be moved is identified, the next issue for optimization, is the determination of the **domain of influence** for the group so that the number of computed grid point locations is minimized. That domain is just the union of the separate domains for the individual points. However, the optimization requires the appropriate care to avoid the potential repetitive calculation due to the overlap of the domains for the control points within the group. Altogether, a strategy is needed to compute the overlap points only once. When the group pattern is rectangular, the domain of influence is readily identified. It is simply given by the minimum and maximum of the respective domains for the associated blending functions (21) in each direction. This is obtained from the functions on the perimeter of the group. In the instances where the group is randomly scattered, the optimization is more complex and was not considered.

The Localization of the Computation for the Initial Grid

When the **CPF** is being applied to generate the grid for the first time, a reduction in the computational effort comes from the **localization of the summations.** The summations are those from the curve generation process which, in the subsequent assembly, were spread throughout the **CPF**. For the computation of any new grid point position, there is a pair of curvilinear coordinate values. Associated with these fixed values are the respective blending functions which do not vanish. It is then clear that there is no need to compute those terms where the blending function is zero. That would simply be wasteful. The consequence, is then to limit the summation. In terms of a single curve (16), the summation is simply given new lower and upper limits which start at the first index associated with a non-zero value and end at the index for the term with the last non-zero value. Altogether, this represents a maximum of three terms for each evaluation. Most of the evaluations occur at this maximum.

With a maximum of three terms for the evaluation of each grid point, there is then a factor of about three times the effort to compute the change in a grid point position by the delta form. In comparison, however, the delta form is applicable only when a previous **CPF** grid is known. In an intuitive sense, two thirds of the data that would have been computed is already contained in the previous grid and is directly utilized in the delta form approach. The computation of the initial grid, by contrast, does not generally have the any previous grid to utilize; and thus, the computation must generally be done from only the control point data along with any arbitrarily specified boundaries. The only notable exception occurs when the attachment process is applied to an arbitrarily given grid and the resulting approximation to that grid is sufficiently accurate to warrant the direct utilization of the given grid in place of the **CPF** grid for the role of the previous grid.

The Localization of the Graphical Operations

The basic interactive graphical template requires the display of the block frame elements, the control nets, the grids, and numerous highlighted positions for user actions. The most consuming of the associated graphical objects are the grids. The reason is because of the much greater detail involved. That is, while all of these basic objects are assembled with various moves and draws, the grids simply involve more of them. Thus, there is a motivation to localize the graphical display of the grids.

The graphical localization of grid display can only be done once there is an object to display the entire grid. Thus, there is no savings in getting the first grid in view. However, once the first grid object is available, the local changes in the grid can be displayed by employing local graphical objects. The procedure is described in the stepwise manner as follows:

(1) Identify the local grid points that have new positions
(2) Create a local object for the old grid points
(3) Change the color to the background
(4) Call the object of (2) to create a hole in the grid
(5) Delete the object of (2) to save storage
(6) Create a local object for the new grid points
(7) Change the color to the grid color
(8) Call the object of (6) to view the changed grid
(9) Go to (1) for further local displays

The tradeoff in efficiency occurs when the size of the global graphical object is twice that of the local one. This is the break even point because the detail in the creation of the global graphical object is the same as that for the creation of two graphical objects of half the size. The two objects are for the creation of the hole in the global grid and then for the filling in of that hole with the object for the new local section. In most instances, however, the local object is substantially smaller than half of the globally one; and accordingly, the associated gain in efficiency is substantial.

While the local graphical scheme has provided a good increase in operational speed, it still needs improvement. In its current form, an additional new graphical object is created in each cycle. That occurs because, in each cycle, two local objects are created; but only one is deleted. As a consequence, the consumed storage due to graphical objects grows as the interactive manipulation progresses. Accordingly, it is reasonable to reset the situation by the regeneration of the global grid after a selected number of cycles. With the reset, the collective local grid objects are simply deleted and replaced by the new global grid object. The point at which a reset is necessary is determined by the amount of available storage for graphical objects. Thus, when the new local grids are generated by the push of a button, the accumulation of the local objects occurs at a reasonably slow rate as to not be much of an inconvenience. When the local manipulation is to be done dynamically in the sense that the new local grid evolves continuously with the control point, the object storage problem must be addressed.

The Prototype Code

The mathematical and graphical schemes for increasing the speed of operation that were developed as described above were also demonstrated in the prototype code. The code was executed on various problems to generate grids and the operational response was seen to be substantially improved over that of the more direct approach in the earlier codes.

The Treatment of Boundaries

The treatment of grid boundaries represents the fundamental means by which the solid objects in a field are described and the individual regional grids are assembled into a multiblock structure. The grid boundaries for the solid objects can be given either in a fixed manner or in a moveable manner which is constrained to lie on the pertinent object. The remaining grid boundaries are transmissive. These are either an unconstrained free boundary or one which is joined to another grid. These latter juncture boundaries are those which are used to glue the individual regional grids together to form a multiblock structure. A reasonable level of continuity at the junctures is typically desired. Without it, the burden of special numerical treatment is then required when the grid is utilized for a simulation. Moreover, when the solid boundaries can be mixed with transmissive boundaries, the number of blocks in a multiblock grid configuration can be greatly reduced. This translates into a more concise and simple grid for a simulation. Altogether, these basic elements of boundary treatment were examined and tested in the prototype code

While the basic treatment for solid boundaries and their manipulation is given by the **CPF** in (61) and (69), the grid spacing and angles at the solid boundary are determined by the adjacent layer of area control points within the block. The substance of this angle and spacing control comes from the curve generation process. In particular, the main control is with those control point curves that enter the given boundary just after passing through the extra layer. The transverse forces are the secondary forces and are represented by the specified boundary, the boundary control point curve, the curve represented by the adjacent layer and the connecting boundaries on either end of the given boundary.. This assumes that there is sufficient distance from the connecting boundaries and that the attachment process in the control point initialization has provided a good control point approximation; for otherwise, there could be a shearing effect arising out of the boundary blending action. The shearing effect would require a compensation before the main action could be enforced. In addition, there is also the assumption that the boundary section under consideration has no slope discontinuities. Such discontinuities do not provide a unique curve tangent direction relative to which an angle can be specified.

With these assumptions, the main action can be witnessed by an examination of a single control point curve such as given in (16). For the single curve, the angle control comes directly from the definition of the vector interpolation in (2) and is executed by the correct alignment of the control point adjacent to the end point in question as in (9). The determination of the spacing of the first grid point from the end requires the derivation of the correct spacing for the adjacent control point. Unlike the case with the angle, the analytical form of the curve does not provide enough.information. The number of grid points , m, is also required. From the first end point, the first increment of arc length, s, is given by the approximation

$$(\Delta s)_1 = \| \mathbf{P}'(\xi_1) \| \Delta\xi \tag{89}$$

where

$$\Delta\xi = \frac{\xi_{N-1} - \xi_1}{m - 1} \tag{90}$$

With the derivative of (10), the increment becomes

$$(\Delta s)_1 = \frac{\psi_1(\xi_1)}{G_1(\xi_{N-1})} \| \mathbf{P}_2 - \mathbf{P}_1 \| \left(\frac{\xi_{N-1} - \xi_1}{m - 1} \right) \tag{91}$$

which, from (11) becomes

$$(\Delta s)_1 = \left(\frac{\xi_{N-1} - \xi_1}{m - 1} \right) \frac{2}{\xi_2 - \xi_1} \| \mathbf{P}_2 - \mathbf{P}_1 \| \tag{92}$$

The first factor in (92) represents the grid point density from parameter space while the remaining part looks like a finite difference. Altogether, with a specified grid point spacing, (92) yields the control point spacing as

$$\| \mathbf{P}_2 - \mathbf{P}_1 \| = \frac{(m-1)(\xi_2 - \xi_1)(\Delta s)_1}{2(\xi_{N-1} - \xi_1)} \qquad (93)$$

which determines the second control point once the angle is known.

The determination of suitable junctures between boundaries can be done in a number of ways. The most basic action is to arrange the area control points on either side of the juncture in such a manner that there is control point alignment at the juncture. Then, when there is also a matching point density along the juncture, the grid is automatically continuous across the juncture. By making the association of pairs of corresponding control points from each block, these are considered to be **tied** together in the sense that they correspond to the same point in physical space. Thus, when a **tied pair** is moved, the motion is a simultaneous motion which keeps the pair as one point. Accordingly, the motion preserves the continuity of the grid across the juncture.

While the use of tied pairs insures grid continuity throughout the motion, it does not provide continuity in either the angles or the spacing across the juncture. The angle and spacing continuity requires the use of the adjacent layers of area control points on either side of the juncture. The earlier tie must then be extended from the pair to include the corresponding control point on each adjacent layer. This **tie of four points** in then arranged in a linear fashion to provide angle continuity. This situation appears as if the juncture were viewed as a solid boundary from each side. Then by giving the inclination of the control point linkage in departing from the juncture, the angle is specified. By matching the linkage angles, the linear arrangement is obtained. This leaves open the possibility to give the angle at will.

With the linear arrangement of four points that is centered about the coincident pair and connects to one point on each side of the pair, the remaining requirement is to determine the spacing for each linkage. To provide a basic level of spacing continuity across the juncture, the grid point spacing was matched on either side. That is, the spacing of (92) on each side is equal. For notation, let the other curvilinear variable be χ, the other number of grid points be n, the other number of control points be L, and the other control points be **Q.** In addition, assume that the juncture corresponds to the first control point on each side. Then the relationship is given by

$$\frac{\| \mathbf{P}_2 - \mathbf{P}_1 \|}{\| \mathbf{Q}_2 - \mathbf{Q}_1 \|} = \left(\frac{m-1}{n-1}\right)\left(\frac{\chi_{L-1} - \chi_1}{\xi_{N-1} - \xi_1}\right)\left(\frac{\xi_2 - \xi_1}{\chi_2 - \chi_1}\right) \qquad (94)$$

which provides the ratio of linkage lengths, but does not give the lengths themselves. Altogether, the spacing from the juncture and the angle through the juncture represent degrees of freedom that can be exercised in an interactive graphical manner. With (94), the choice of one linkage length is enough to determine the spacing; and with the inclination of the linear arrangement, the angle is determined. In an operative sense, the linkage lengths fix the linear configuration of the tied points while the rigid motion of the configuration determines the rest. That includes both translation and rotation.

The next step beyond the matching of grid point spacing and angles at the juncture is to account for the rate of growth for the spacing and angles which pass through the juncture. The consequence is a modification of the above prescription. For the linkages, (94) is changed to include a growth rate factor on the right hand side. As it is stated, the growth rate is unity which means that the spacing on each side is equal. With angle growth rates, the linear configuration is changed into a modest bend. Altogether, the growth rates represent a higher order effect and thus a refinement. In the execution of a dynamic manipulation, the mechanics of reconfiguring the tie adds an additional level of complexity. In the code the accounting for growth rates were not undertaken.

The case of junctures with aligned area control points provides the most readily applicable means for locally modeling. While the modeling action is more direct with the alignment which permits the above four point ties, the initialization of the block structure of control points requires more effort. Thus, there is a balance between the places where the effort is expended. For this balancing option to be available, the means to treat junctures must be established for the case where there is no alignment of the area control points that come from each side. Several strategies have been established for this purpose.

The **first strategy** is to represent the juncture by a separate sequence of edge control points and to then employ the bottoms-up self-consistent formulation of the **CPF** as given in (69). The constraint upon the edge control points is that the generated edge grid points are reasonably close to the corresponding edge representations that would come from the area control points on either side. This is separated into a closeness in the geometric shape and in the distribution of the grid points. The distribution is, perhaps, less intuitive, but is also very important because a disparity there would enter the boundary blending terms of (69) and would result in a distortion of angles. The consequence of a greatly misaligned pointwise distribution would be the requirement for a rather non-intuitive compensating adjustment of the adjacent layers of area control points.

With the motivation to select a suitable sequence of edge control points for the juncture in the first strategy, the existing area control points that come from either side are considered. They certainly are close in the geometric sense. To most readily address the pointwise distribution, it is most convenient to choose it from only one side. This alleviates the need to carefully choose appropriate points of interpolation (4). For the most flexibility, the side with the richest supply of area control points is preferred. Typically, the richest supply comes from small coordinate blocks that are appendages to a larger block. As a secondary problem, the corner treatment must also be done carefully.

Altogether, the entire interface curve represents a **tie** which glues the two blocks together with continuity. The continuation, as with the earlier transition from pair ties to four point ties, the adjacent layers are then included in the tie operation. The objective, as before, is to enforce continuity in the grid spacing and angles through the juncture. While the basic idea is the same, the pattern of these ties is more complex and varied.

In conjunction with the above first strategy, a **second strategy** is needed to obtain spacing and angle continuity. This was done by employing an overlaid area control net. This third area control net is designed to cover the juncture as an interior curve and to be an essentially local net. The sequence of operations is to first attach the overlaid net, then to generate the local grid which includes the juncture in its interior, to next attach the area control nets on either side, and to then generate the grids on either side by using the one sided approach of the first strategy, but not with any special ties for the adjacent layers. In a direct manner, the configuration of the adjacent layers are automatically determined by the attachment process. In continuation, further means for adjacent layer determination can be expected to arise and be best in for certain actions. Moreover, the extension into additional layers can be expected to evolve. Such continuations and extensions were not pursued.

In situations when a block side contains a mixture of solid and transmissive boundaries, there must be a means to make a smooth transition occur between the solid and transmissive sections. The transition is accomplished by extending the **CPF** by replacing the switches between specified and free-form boundaries. Those switches appeared in (61) and (69) with a value of 1 for the specified case and a value of 0 for the free-form case. While these constant values are quite sufficient for boundaries of a single type, they are not for one of mixed type. The straight forward assignment of 1 and 0 for specified and free-form would lead to a discontinuity at any point of transition. In effect, the specified side would have a non zero boundary blending term and the free-form side would not. The disparity at the transition point would then propagate into the grid by a distance given by the support of the blending function (21). To prevent the discontinuity arising from this disparity, the otherwise piecewise constant function with values of 0 and 1 is altered to provide smooth transitions between 0 and 1. For each such transition, the function value at the point of transition is 1 because of the specification requirement. Then, as the free-form section is entered, the function smoothly decays to 0. As a free-form region is left, the function simply performs in the reverse fashion by increasing from 0 to a value of 1. Each transitional step was given by domain scalings, translations, and reflections of the simple step function

$$f(w) = w^2 (3 - 2w) \qquad \text{for} \qquad 0 \le w \le 1 \tag{95}$$

where the domain position and orientation appears in w. This function clearly converts the piecewise constant switching function into a derivative continuous switching function. Thus, accordingly, the constant switches ρ_k of (61) and (69) are replaced with these functions.

Automatic Features

The automatic features that were considered are for orthogonality and spacing along a boundary segment, rubber banding and sheeting, and global elliptic schemes.

The **orthogonality** and **spacing** along the boundaries is achieved to a reasonable level by adjusting the area control net linkages from a boundary to be orthogonal and to have lengths determined by (93). These linkage determinations can be done separately. The grid spacing from the boundary can be given while keeping the current grid angles; and conversely, the grid orthogonality can be given while keeping the current grid spacing. In each instance, the relocation of the control points, adjacent to the boundary under consideration, is an approximate solution to a variational problem. This is due to the fact that there are more grid points than area control points along the boundary. Accordingly, an overdetermined situation has arisen when control point movement is employed to meet the grid requirement for orthogonality and/or spacing. An analytical statement of the variational problem is not unique. In a direct sense, it can be taken to be the minimization of a linear combination of two integrals. The integrand of the first is the cross metric or some multiple of it. The second integrand is something like the squared difference between the specified transverse metric and its desired prescription. The square root of the transverse metric and its prescription can also be employed, but is more complex although closer to the spacing itself. To improve the approximation, an iterative scheme is natural and can be stated in an analytical form. In a good number of cases, however, the simple direct movement of each control point on an individual intuitive basis does provide a fairly good approximation. Only this intuitive step was taken in the prototype code.

In the case of **global elliptic schemes**, only the general set up was considered. The essential strategy set fourth was to choose a sparse control net and have the initial associated interpolation functions of (11), (12), and (13) be defined over a uniform partition (4). Then, by uniformity (30), the starting control point configuration has half point spacing from the boundaries and full point spacing in the interior. By the injection of an artificial point at the midpoint of each interior interval, a sparse grid is obtained for which each the control net appears in an embedded form. With the attachment process built in form the start, the movement as a coordinate transformation preserves the attachment. Then with the application of any standard elliptic scheme to the sparse grid so determined; the movement process is exceedingly efficient because of the small number of (provisional) grid points. Once the movement is completed, the control points are extracted and the **CPF** is then employed to generate the grid of the desired density.

The strategies for **rubber banding** and **sheeting** represent exceedingly useful tools. They are used when a given control point is to be moved a substantial distance and would otherwise over run or greatly separate from neighboring control points. Thus, the neighbors are moved along a rubber band or sheet which is hinged to a bounding section of control points and tied to the current control point at the center of the movement action. For curves, the operation is called **rubber banding** and the hinge points appear on either side of the center point. The action in the prototype code was with linear connects to the hinges and a progressive tracking of those connects. The **rubber sheeting** is the two dimensional parallel. The neighboring points were taken to be a rectangular control point section for which the action point appeared in the center. The hinges were taken to be the control points on the perimeter of the section. The formula for the movement was transfinite in spirit, but was constructed to conform to the action point. With the assumption of a modestly distorted neighborhood, a rubber sheeting function was constructed in parameter space with linear connects from a value of 1 at the action point to values of 0 at each surrounding hinge point. Altogether, the sheeting function appeared as a tent-like function The values at each control point position in parameter space were then employed to appropriately down scale the movement of the action point center for each of the neighboring control points. When the neighborhood is more distorted, an augmentation to a full transfinite interpolation procedure would be required. The simplest scheme of this sort is to use the previous control points to define a discrete mapping; then, to move the points first in parameter space; and finally, to map ;the results into physical space. In the prototype program only the first and most direct approach to rubber sheeting was considered. Improvements and refinements were left for Phase II.

The Prototype Code

The **initialization** procedure that was established contains an extension of the control point **attachment** process, several **restarting** options and the **self-consistent** treatment of boundaries. The **self-consistent** treatment of bounoaries required the creation of schemes for storage allocation and for the **assembly** of the basic structural components. The consistency stems from the utilization of the control point transformation for both the boundaries and the interior regions. Thus, the transformation was programed generally enough to apply to both the block edges and areas. The **assembly** was done in a manner which insured that the various blocks fit together with the requisite continuity to start the control point manipulation process of the other tasks. The continuity requirement was satisfied by utilizing the basic theoretic formulation. A simple illustration is given by the treatment of block edges which are to be transversely attached to other edges. To start from a given edge, the transverse edge must indeed leave from a grid point on the actual curve. The need was to identify the grid point and to put it in relationship with the adjacent control points for the block interiors. To reflect the self-consistent structuring, the **attachment** process was done for both boundary edges and the interior block areas. This has given two options: (1) a representation entirely in terms of control points and (2) a representation where arbitrarily specified boundaries are present. The **restarts** were also created with two options: (1) a direct restart from the constructive data elements of the formulation by employing only a read statement and (2) an attachment to a given block structured grid.

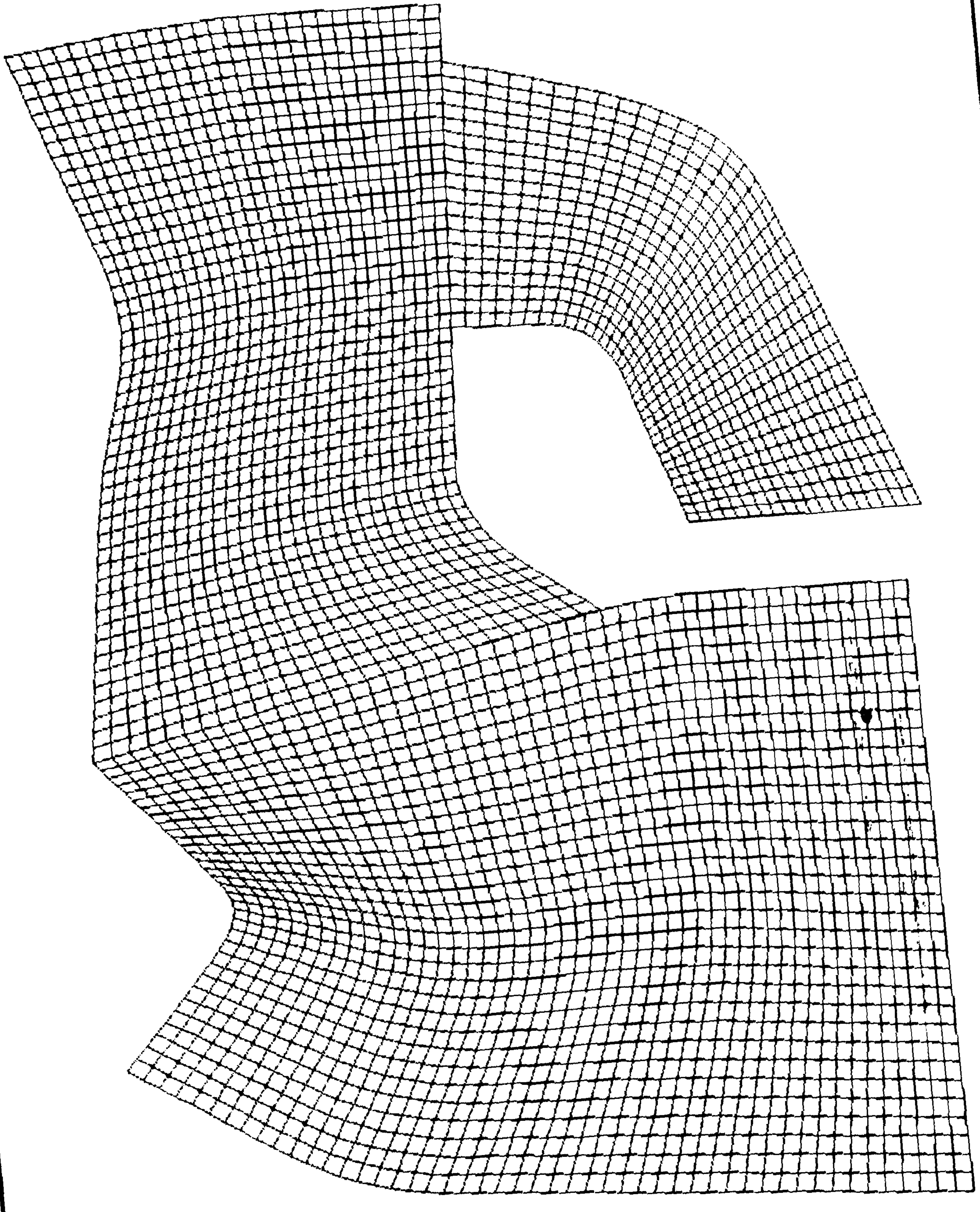
The establishment of **grid topology** was accomplished within the context of multiple blocks and control points. The compactness of the control point representation was utilized. That utilization was most prominently displayed in the sequence of operations. This was done in a bottoms up manner. First, corner points are set, then linear connects are given between them, edge control nets are next attached to the linear segments, and then the control points are interactively moved to deform the edge boundaries into the desired shapes. The result is the multiblock frame that represents the collective boundaries. Then (with the frame) the bottoms up approach is completed when the areas are filled in with area control nets and associated grids. The more technical aspects involve the ordering of operations, the mathematical interrelationships of the various components, the software development to execute the mathematical requirements, and the techniques of interactive graphics.

The **boundary treatment** was established for **junctures** between blocks and for **mixed conditions** on edges. The treatment involved the ability to locally manipulate the boundaries in a manner which automatically maintains a good level grid quality. The quality, here, is to maintain pointwise continuity, continuity in spacing and continuity in angles. The procedures established for **junctures** were based upon setting fourth a local configuration of control points that then were moved in a rigid fashion. The configuration employed a control point layer on either side of the juncture. The points on either side were positioned in such a manner as to insure the desired grid quality. The configuration was then tied together so that it could be move as a rigid unit. This was done with the most basic configuration which is the linear arrangement transversely piercing the juncture. The **mixed conditions** were established by applying a blending function (95) for the boundary conformity terms of the control point transformation. The effect of the new blending action was to smoothly transition from a precise boundary specification into a free-form representation that can be actively manipulated as if it were an ordinary juncture.

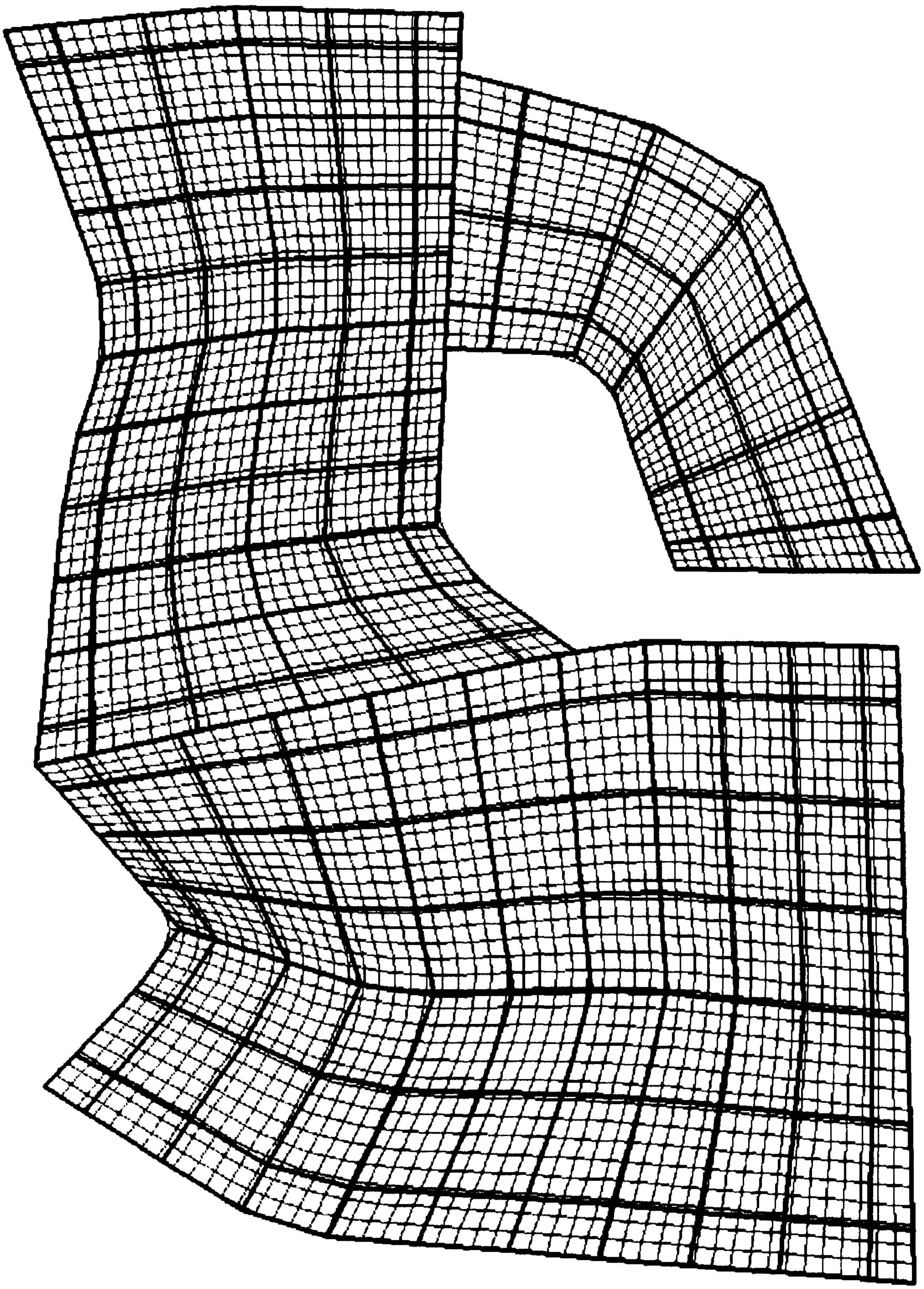
The **speed** for interactive response time was greatly enhanced so that the rapid application on small personal computers is clearly viable and would in fact be quite impressive (the current prototype runs on an IRIS 2500 workstation). The speed was examined for the manipulation of one control point. The results appeared virtually instantly with the press of a key (the P key in the code). The elements of the accelerated computation were the localization of the arithmetic operations, the localization of the graphical operations, the sequencing of operations, the balancing between storage and computations, and the fast evaluation of the transformation from its internal structure.

The **automatic features** that were established in the code included boundary orthogonality, rubber banding, rubber sheeting, and boundary spacing. The orthogonality was accomplished by the redistribution of the adjacent layer of control points. The rubber banding was done on sequences of control points that were bracketed by a hinge point at one end and a moveable point for the other end. When the moveable end undergoes a displacement, the collection along the band also does. This was demonstrated in a progressive manner in which the band need not be immediately a linear arrangement. Rubber sheeting was done in the same manner but was more complex. The hinge points were replaced by a surrounding set of control points. The boundary spacing was handled by using point densities.

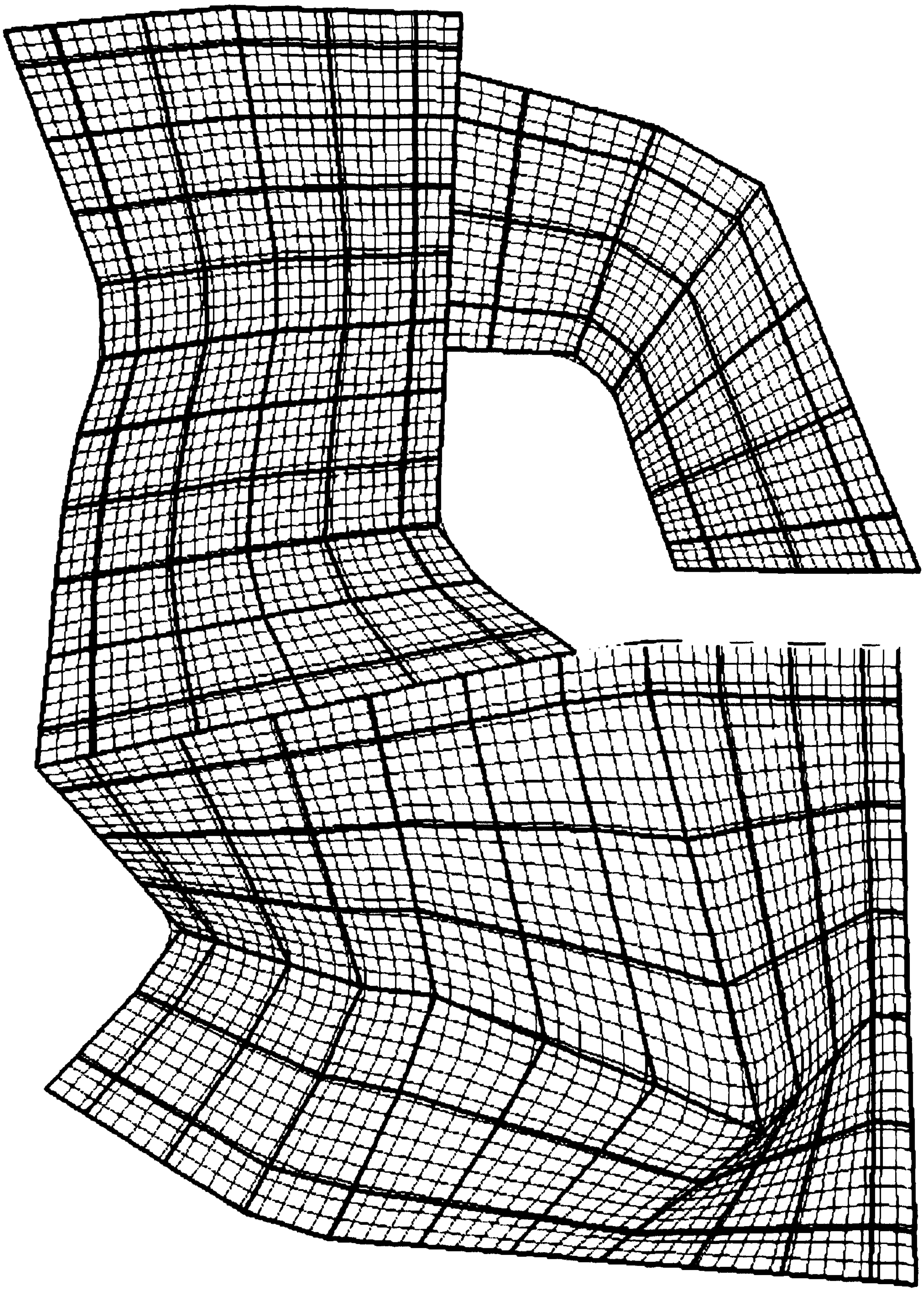
Altogether, the prototype code was written in about 2000 lines of FORTRAN 77. Actually this was the second of two prototypes. The first one, however, was restricted to a single 2D block, but was employed to more simply experiment with the ehancement of the speed of operation. It was about 600 lines of FORTRAN 77. The speed enhancement strategy was then incorporated into the multiblock prototype. For greater flexibility, the codes to be established in Phase II and III would utilize C and C++. The flexibility would come from the greater ease of treating general data structures and the greater modularization that comes with an object oriented approach. Some plots from the code are displayed below. They show a basic rubber sheeting and juncture manipulation operation.



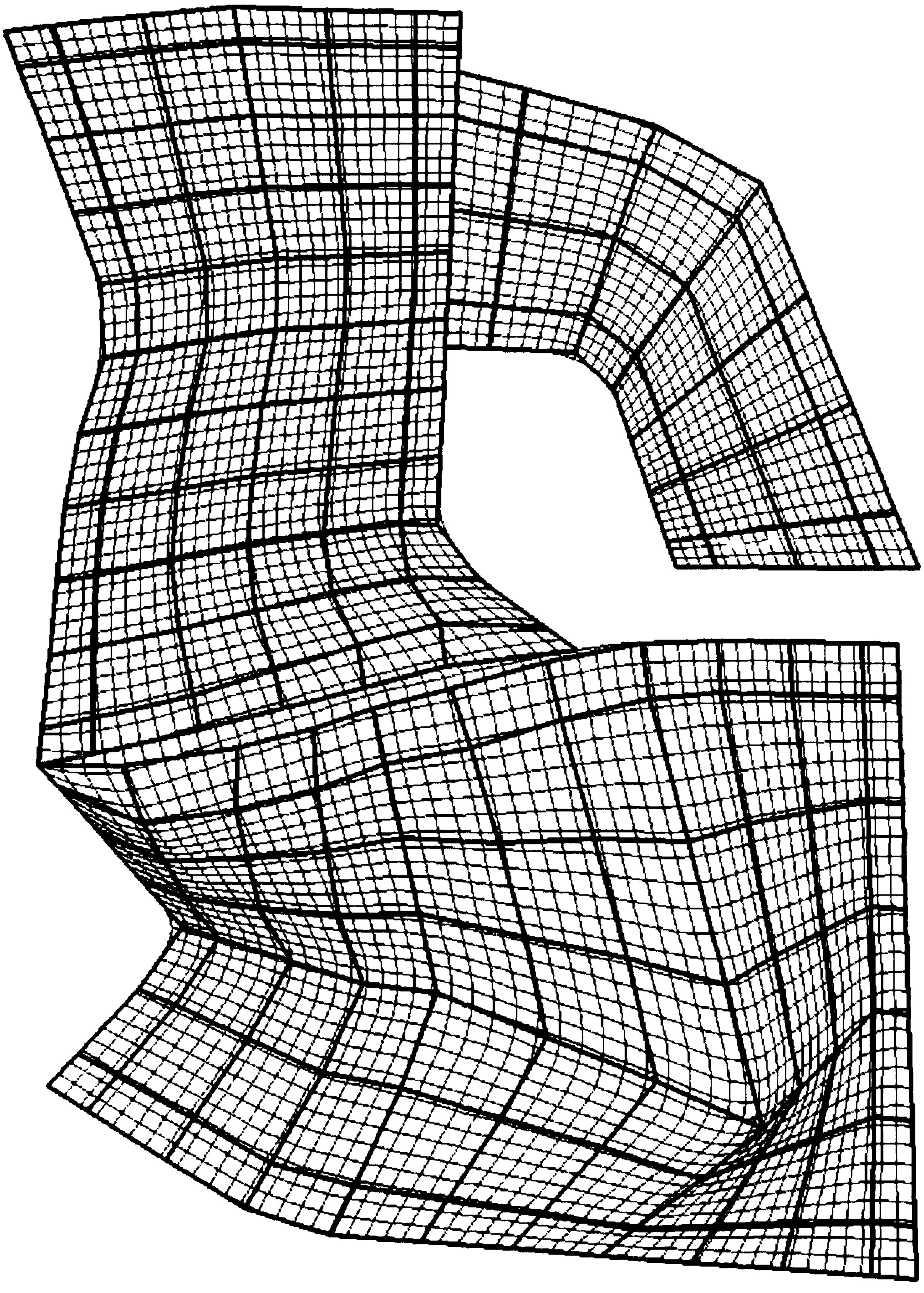
Initial Grid



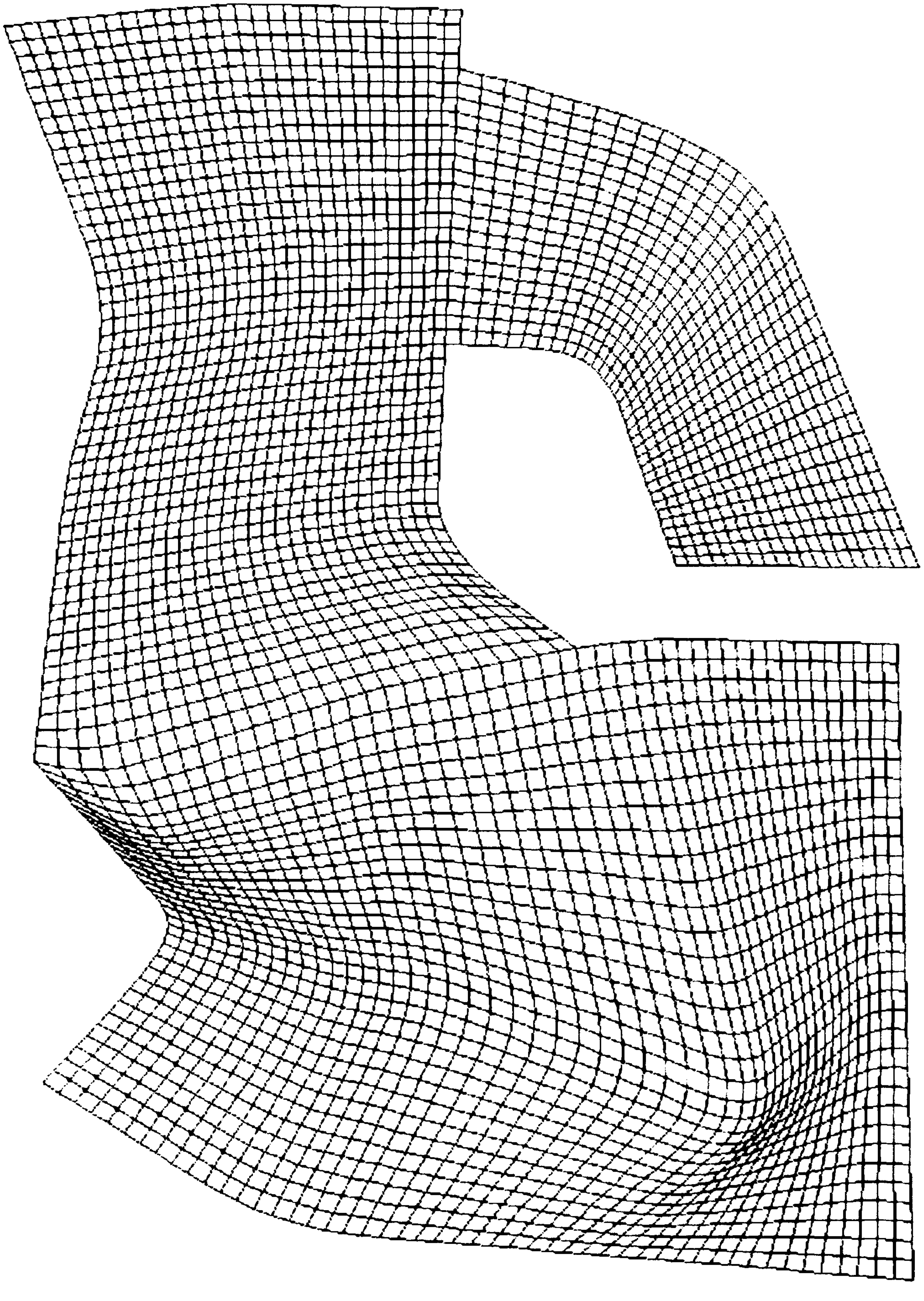
Initial Attached Non-Aligned Control Net



Rubber Sheeting Operation



Juncture Manipulation With Non-Conforming Net Interface



Grid With Both Rubber Sheeting and Modestly Altered Juncture